Manual

Inhoud

[1 Webscraping 2](#_Toc88697570)

[1.1 Websites 2](#_Toc88697571)

[1.2 Tools 2](#_Toc88697572)

[2 Docker 3](#_Toc88697573)

[2.1 Terms 3](#_Toc88697574)

[2.2 Installation 3](#_Toc88697589)

[3 Git 5](#_Toc88697590)

[4 Neo4j 6](#_Toc88697591)

[4.1 Download links 6](#_Toc88697592)

[4.2 Run neo4j 6](#_Toc88697593)

[4.3 Neo4jdriver 6](#_Toc88697594)

[4.4 The neo4jdriver is a tool that helps you store your data into a neo4j database. 6](#_Toc88697595)

[4.5 Cypher 7](#_Toc88697596)

[5 Schema of the process 8](#_Toc88697597)

# Webscraping

Webscraping is used to gather the data from websites. This data can be used in the next steps.

## Websites

To webscrape websites is it necessary that it is know which websites to webscrape. For this project there was an Excel file made with all the necessary websites. This file includes the name of the company, URL and if it is a Dutch or English website. The last one is important for the uses of domains in the NLP part of this project. With this Excel file can the websites easily be webscraped.

So:

1. Determine the websites that need to be scraped and put them in an Excel file. (name and URL)
2. Put in a separate column the language that is majority used on the website.

When these steps are taken, the following can result:



## Tools

For webscraping the websites, the following tools are used:

* Pycharm
* Python library BeautifulSoup

Pycharm is an integrated development environment used to program in python. To set up Pycharm do the following:

1. Go to the following website and download Pycharm: <https://www.jetbrains.com/pycharm/download/#section=windows>
2. Follow the installation instructions.
3. Make a new project and file.

For the webscraping there are several libraries used such pandas and request. For the webscraping in particular, BeautifulSoup was used.

1. Install the required libraries.
2. Code!

There is a lot of documentation on BeautifulSoup to use when creating the webscraper.

# Docker

Docker is an open-source service that allows users to package an application with all its dependencies into a so-called container that can run on any Linux, Windows, or macOS computer. Unlike virtual machines, containers do not have high overhead and hence enable more efficient usage of the underlying system and resources.

## Terms

|  |  |
| --- | --- |
| Term | Description |
| DockerFile | * instructions for how to build the Docker container image
* automates the process of Docker image creation
* a list of command-line interface (CLI) instructions that Docker Engine will run in order to assemble the image
 |
| Docker images | * contain executable application source code as well as all the tools, libraries, and dependencies that the application code needs to run as a container
* when you run the Docker image, it becomes one instance (or multiple instances) of the container
 |
| Docker containers | * live, running instances of Docker images
* while Docker images are read-only files, containers are live, ephemeral, executable content
* users can interact with them, and administrators can adjust their settings and conditions using docker commands
 |
| Docker Hub | * public repository of Docker images
 |
| Docker daemon | * service running operating system, such as Microsoft Windows or Apple MacOS or iOS
* creates and manages your Docker images for you using the commands from the client, acting as the control center of your Docker implementation.
 |
| Docker registry | * scalable open-source storage and distribution system for docker images
* enables to track image versions in repositories, using tagging for identification
 |

## Installation

<https://docs.docker.com/get-docker/> - for all platforms

To launch *test container*:

|  |  |  |
| --- | --- | --- |
| Step | Where | What to do |
| 1 | List of applications | Open Docker Desktop  | - |
| 2 | Terminal / Command line | Run command | docker run -d -p 80:80 docker/getting-started |
| 3 | Browser | Open link | <http://localhost:80> |

‘Getting started’ page should be opened. In Docker desktop application you will see running container.

*EyeOnText docker installation:*

|  |  |  |
| --- | --- | --- |
| Step | Where | What to do |
| 1 | List of applications | Open Docker Desktop  | - |
| 2 | Terminal / Command line | Runcommand | docker login <https://docker.eyeontext.com> -u partners |
| 3 | Terminal / Command line | Enter password | You will be asked for a password, which you can request via info@eyeontext.com |
| 4 | Terminal / Command line | Run command | **docker -run -v [folder]:/eotws/ -it docker.eyeontext.com/eot/sdk:2.1.0 bash***NOTE: CHANGE ‘[folder] TO THE FULL PATH OF THE FOLDER YOU WANT TO SHARE FROM YOUR LOCAL COMPUTER* |
| 5 | Terminal / Command line | Run command | **docker run –p 5000:5000 –it docker.eyeontext.com/eot/sdk:2.1.0 demo\_server***NOTE: YOU NEED TO RUN THIS COMMAND IN NEW TERMINAL WINDOW* |
| 6 | Browser | Open link | <http://localhost:5000/wowool/playground/> |
| 7 | Browser | Open link | <http://localhost:5000/wowool/playground/static/tir/docs/index.html> |

A new container will appear in Docker Desktop application. Now you can start/stop this demo from Docker application (so no need to run code above again).



# Git

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Just as Google Drive makes it possible for multiple contributors to write, edit and add to the contents of a single text file, git is a computer program making it possible for multiple coders (and project managers, testers, content providers, and whomever else is on the team) to collaborate on a single project.

This provides a beneficial working environment for the team and helps everyone work on the same issues while having control of the versions of the project, clear view of what changes were made, and when, and provides insight into how the development was made, who made them and, if documented properly, with descriptive comments tagged alongside the changes made.

**Github Desktop** is an application that enables you to interact with GitHub using a GUI instead of the command line or a web browser. ... You can push to, pull from, and clone remote repositories with GitHub Desktop, and use collaborative tools such as attributing commits and creating pull requests.

Installation guide:

1. Visit <https://desktop.github.com/>

2. Click the “Download for Windows” button

3. In your computer’s Downloads folder, double-click the GitHub Desktop setup file.

4. GitHub Desktop will launch after installation is complete

# Neo4j

Neo4j graph database is the most trusted, secure, and globally deployed graph database. At real-world scale, you get no-compromise performance, reliability, and data integrity. Neo4j is the only enterprise-strength graph database that combines native graph storage, scalable speed-optimized architecture, and ACID compliance. The high-performance distributed cluster architecture of Neo4j enables the most challenging OLTP and data science workloads.

## Download links

In MacOs/Linux:

And download the neo4j community edition:

<https://go.neo4j.com/download-thanks.html?edition=community&release=4.3.6&flavour=winzip>

In Windows:

download neo4j:

<https://go.neo4j.com/download-thanks.html?edition=community&release=4.3.6&flavour=winzip>

download jre: (java)

<https://builds.openlogic.com/downloadJDK/openlogic-openjdk-jre/11.0.8%2B10/openlogic-openjdk-jre-11.0.8%2B10-windows-x64.zip>

## Run neo4j

1. Go to the location where you have extracted neo4j and run the server
2. Type following:

bin\neo4j.bat console

Or

./neo4j console

1. To check if server is running go to:  <http://localhost:7474/browser/>
2. Create a test user with test as a password from your neo4j console:

Create or replace user test

Set password ‘test’

Change not required

## Neo4jdriver

## The neo4jdriver is a tool that helps you store your data into a neo4j database.

From the docker container type:

Neo4jdriver.native –help

Running neo4jdriver in docker (Adjust the path accordingly)

neo4jdriver.native -l english -f /eotws/corpus/digireal -x /eotws/corpus/digireal/digireal.download --domain dutch-entity,/eotws/rules -s <http://host.docker.internal:7474>

Syntax explanation

-f is the location of the data:

-l the language code

-x the json link file.

--domain a comma delimited list of domains.

  Note that a domain can be a folder name with wowoolian file in it.

-n Perform a dryrun, this will display the cypher query on screen but not execute it.

--topics

So we need 3 elements:

1. A corpus of texts (in this case we have used  movie plots)
2. A domain, could be existing ones, but it is more interesting if you create your own, as we have done with the movies
3. A link file (.lnk) , that explains which expressions to use as nodes in the database and how they can be connected (have a look at the one at: test/projects/movies/movies.lnk)

Check the results in your browser (<http://localhost:7474/browser/>) by typing the following **CYPHER** query (query language of neo4j):

 match (n) return n

## Cypher

Cypher is powerful, intuitive, graph-optimized query language that understands, and takes advantage of, data connections. It’s user-friendly, easy to learn, and follows connections – in any directions – to reveal previously unknown relationships and clusters.

When trying to find patterns or insights within data, Cypher queries are much simpler and easier to write than massive SQL joins. Since Neo4j doesn’t have tables, there are no joins to worry about. Compare the Cypher query at the left its equivalent in SQL.

* Check contents of database

MATCH (n) RETURN n

* Matching a particular annotation

MATCH (n1:Person)-[r]-(n2) RETURN r, n1, n2 LIMIT 250

* Matching 2 Annotations

MATCH (n1)-[r]-(n2)

WHERE n1:Education or n1:Innovation

RETURN r, n1,n2 LIMIT 25

* Matching a string

MATCH (n1)-[r]-(n2) WHERE n1.name = ‘talent’ RETURN r, n1,n2 LIMIT 250

* Matching a topics

MATCH (n1:topic)-[r]-(n2)-[r1]-(n3) RETURN r, r1,n1,n2,n3 LIMIT 250

* To delete database

MATCH (n) DETACH DELETE n

# Schema of the process







## NLP

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software.

NLP: interaction between computer - human

* Speech recognition
* Language generation
* Machine translation
* Natural language understanding

**Wowool language**

 This language already has pre-installed libraries with different languages (Add list of languages) and domains (add list and explain)

Wowool is a text-oriented pattern matcher engine. It has two main components:

**Lexicons (Like vocabulary)**

Easiest way to annotate, just matching literal or stems; add knowledge of the world, simplify rule writing.

Example: In our case we had a problem that NLP was detecting words (Engels,
 manage, As, ways, Sophia Rotterdam) as city. Datastic had to filter out those words and let the program know that they are not cities ( = NoCity)



(input="normalized\_literal") - Matches the literal token regardless of capitalization or any accents or special characters.

Lexicon for Identifying different phone formats from web sites. The problem is that every website uses its own format, but they are limited.



* ([:digit:]){9} - if there is 9 digits - it is a Telephone number (Random example: 061234567).
* (.)\* - space between numbers.

**Rules (Like grammar)**

In comparison to lexicons, rules provide a more expressive tool for capturing linguistic patterns. The use of lexicons in the creation of dictionaries, gazetters, and rules in the creation of larger and more complex patterns:

With rules you can:

* address stems, literals and part of speech, as done in lexicons
* use other annotations (from lexicons or other rules)
* Address the context of the expression
* Make sub-annotations
* Use the relative position of the sentence
* Make coreference rules
* Use filters
* call python

**Example:**

To identify what kind of technologies the given text has, to extract this data and mark it as “Technology”. The rule says that if there are more than 0 words and less or equal 3 before word “Device” and those words are Noun or Adjective or Adverb – this combination of words is Technology.

 

**Link file**

Link file used to set a connection between needed topics and create visual relationship between them.

Example: Datastic team would like to see what kind of projects every university has and how do they cross with each other based on topics.

 

This code set the connection between uploaded file to scan (“Doc”) and names on university projects with “Projects” label. But to see what those projects are about there must be created one more connection between project names and topics. This way will help to see if different universities have same project topics that they could potentially work on together.

**Domain**

A domain is a collection of wow files (rules and lexicons) that are grouped for a single purpose (healthcare, finance, recruitment). Those files work as a library to extract needed data from txt files based on topic required to identify.

### Environment

 File storing solution

All files and folders are stored in “eotws” folder:

 

Files to scan:

 



 Rules for NLP are adjusted and stored in subjects.wow file.



 Links that create connection between different topics.

